Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 29 (2012) 1154 – 1159

Procedia Engineering

www.elsevier.com/locate/procedia

2012 International Workshop on Information and Electronics Engineering (IWIEE)

Dendriform Branch Cut Algorithm Based on Minimum Spanning Tree for Phase Unwrapping

Zhang Sen[a]*, Zhong Heping[a], Tang Jinsong[a]

[a]*Electronics Eng. College, Naval Univ. of Engineering, Wuhan, Hubei, China*

Abstract

Two-dimensional (2D) phase unwrapping is a key data-processing step of interferometric synthetic aperture radar (InSAR). After analyzing the classical branch cut algorithm, the method of first-child, next-sibling is adopted to express the residues connective relationship, and a dendriform branch cut algorithm is proposed which can remove the 'dead area' existing in the classical branch cut algorithm. A branch cut regrowing strategy is proposed to improve the simple dendriform branch cut algorithm, which keeps the high efficiency of the original algorithm and solves the discontinuous problem caused by the long branch cuts connection. The experiment result using real InSAR data indicates that the proposed method is effective, and greatly improves the phase unwrapping result.

Keywords: InSAR; Phase unwrapping; Branch-cut regrowing; Minimum spanning tree

1. Introduction

The phase difference deduced from a pair of accurately registered complex image will be wrapped into the interval $[-\pi,\pi)$, which is the phase principal value of the real phase. The process that we get the real phase difference by adding integral multiple of 2π to the wrapped phase is phase unwrapping. 2D phase unwrapping has received a great deal of attention because it's widely used in synthetic aperture radar (SAR) and synthetic aperture sonar (SAS). Consistency and accuracy are the two main problems that should consider mostly during the process of phase unwrapping. In practice, the process of phase unwrapping becomes quite difficult for low coherence, abrupt elevation change of the imaging areas.

* Corresponding author. Tel.: +86-27-83444036;.
E-mail address: Johnson_xh@sina.com.

doi:10.1016/j.proeng.2012.01.104

Nowadays, the existing phase unwrapping methods have been divided into three groups: path-following methods [1, 2], minimum-norm methods [3, 4] and statistical methods based on maximum posterior estimation [5] or maximum likelihood estimation [6]. Path-following methods keep the local optimization and accuracy, but the unwrapping phase may get abruptly 2π jump, while the minimum-norm methods hold the global optimization and lost the accuracy due to over-smoothing. In 1986, the branch cut algorithm proposed by Goldstein is the most known path-following methods. For its high efficiency, the accuracy of the unwrapping result of the wrapped phase with low noise and few residues by branch cut algorithm is very perfect. But when the residues become densely, the algorithm fails to connect the branch cut correctly due to the nearest-neighbor strategy. The minimum spanning tree method proposed in [7] is a good attempt, which selects the geometric center of the current spanning tree as the reference point and adds the new nearest residues not in the tree into the current spanning tree until it becomes neutral. During the process of spanning, the newly added nearest residue to the geometric center of the current spanning tree may be not the nearest residue to the current spanning tree, so the branch cut may be optimized further.

In this paper, a new branch cut generation method and its regrowing strategy is proposed. During the process of branch cut generation, we add the nearest residues into the current spanning tree until it becomes neutral. This new method reduces the length of branch cut dramatically, removes the ‘dead areas’ formed by the classic branch cut method and optimizes the unwrapping result. Finally, the real InSAR dataset is used to validate the performance of the proposed method.

2. Principle of branch cut generation

2.1. Residues theory

Itoh’s [8] method is a specific implementation of the notation that the unwrapped phase can be obtained by integration of the phase gradient. This concept can be extended to two dimensional signals with the following representation. Assuming the phase gradients are known along with the phase at some initial point r_0, the phase at some other point r is obtained from the following path integral:

$$\phi(r) = \int_C \nabla\phi dr + \phi(r_0)$$

where C is any path in 2D space connecting the points r_0 and r, and $\nabla\phi$ is the phase gradient. From calculus, the integral will be path independent when the phase gradient $\nabla\phi$ happens to be an exact differential over the domain of interest or it is an irrotational field. But most of the time, wrapped phase fields will violate the conditions for path independent owning to noise, under-sampling, overlapping and so on, therefore the phase unwrapping becomes to select a path covers the domain that simultaneously satisfied certain properties which ensures the unwrapping result perfect. Fortunately, it is clear that this inconsistency is only restricted in isolated points and regions, and which can be easily detected. Ghiglia and Pritt [9] constructed the residue theorem for 2D phase unwrapping, just as in complex variable contour integration. Therefore

$$\oint_C \nabla\phi dr = 2\pi \cdot \text{(sum of the enclosed residue charges)}.$$

In other words, the closed path integral around a phase residue will equal some integer multiple of 2π, so a consistent phase unwrapping is possible if and only if all integration paths do not encircle unbalanced residues, which is the essence of branch cut algorithm. While in practice, except the positive and negative residues must be balanced, some other constraint conditions should be imposed in order to put the branch cuts more correctly. The ‘dead area’ in the unwrapping domain will appear due to improper branch cut setting, and long branch cut will make the unwrapping result discontinuous. The branch cuts setting and optimization become the most important problem in the branch cut algorithm.

2.2. Branch cut based on spanning tree

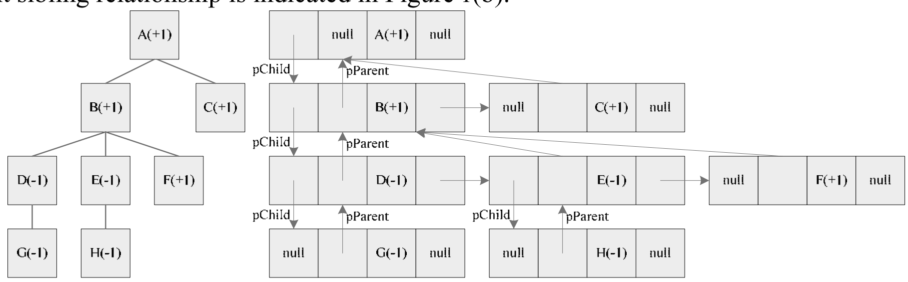
A tree T we defined is a finite set composed by $n(n>0)$ nodes, which should satisfy the following two conditions: (1)Have one and only one root node; (2)The other nodes are divided into $m(m \geq 0)$ groups T_0, T_1, $\cdots$, T_{m-1}, which are not intersectant. Every group is a new tree and a subtree of the root node.

Tree is a graph without cycles, which ensures the 'dead area' will not appear for improper setting of branch cut after organizing the residues by tree structure. During the process of spanning, when the residues have not balanced, we continually add the nearest residues to the current spanning tree and connect it with the nearest residue already on the tree until the current tree gets balanced. But the sole terminal condition by balancing residues will make some branch cuts spanning too long and the unwrapping result discontinuous. A re-optimization method for phase unwrapping based on minimum spanning tree is also proposed, during the spanning process, a given balanced tree may be re-activated several times and regrow again.

3. Algorithm implementation

3.1. Data structure design

Considered one residue may be connected with more than one residues and the number of the residues to be connected is not a constant, the first child, next sibling relationship representation is selected to store the connective relationship of the residues. The standard first child, next sibling relationship representation has low efficiency in finding its parent node from one node which must start from the root and search one by one. In order to generate the branch cuts quickly from the connective relationship, we add a new field in the standard fields, which is used to conserve the current node's parent node by a point. For the residues connection shown in Figure 1(a), the corresponding storing structure using the first child, next sibling relationship is indicated in Figure 1(b).



(a) (b)

Figure 1. Residues connection and its optimized first child, next sibling storing structure

3.2. Simple minimum spanning tree algorithm (SMSTA)

The simple minimum spanning tree in this paper modifies Goldstein's algorithm [9] in two aspects: (1) Unbalanced residues don't connect to the already balanced residue; (2) The connective relationship among the residues stores according to the first child, next sibling relationship. The charge is balanced by connecting residues using minimum spanning tree, and we can get an ideal unwrapping result by the simple method in most cases. The detailed implementation is following:

Step1：Firstly we initialize the sole root node of the global tree and the sole ground node, then initialize the residues nodes according to the residues array identified by the wrapped phase.

Step2：Scanning the residues array from top to bottom and left to right, if unvisited residue is found, we change its label as visited and add it to the global tree as its child, then we regard the unvisited residue node as the root of one subtree and the charge of the current residue as the cumulative charge of the current tree; if unvisited residue is not found, we jump to Step5.

Step3：Traversing the subtree according to the post-order searching algorithm to find the node with the nearest residue by the information we computed and stored in the node's fields. If the nearest residue is visited, we re-compute the nearest unvisited node of the node we found and jump to Step3, else we add the nearest node to its nearest node in the tree as its child; If the nearest node is the ground node, which means that the branch cuts have spanned over, we set the cumulative charge of current tree to zero and jump to Step2, else we add the charge of the nearest node to the cumulative charge and re-compute the nearest unvisited node of the two we just operated on.

Step4：If the cumulative charge of the subtree is equal to zero, then jump to Step2, else jump to Step3.

Step5：After finishing spanning, we traverse the all children's node of the global root node by pre-order algorithm and set the branch cuts.

During the spanning process, the charge is balanced only by simply connecting residues. Using the simple spanning tree algorithm, the topology of a tree can't be changed after it's balanced. The branch cuts shown in Figure 2(a) are connected by the classic branch cut algorithm. There exist lots of redundancies of branch cuts because a residue will be connected to the all nearest residues until the current tree is balanced. The branch cuts shown in Figure 2(b) are connected by the simple spanning tree algorithm, compared with Figure 2(a), two branch cuts are removed by one residue only connecting to the nearest unvisited residues. But we can see that the branch cuts still are long and get intersectant from Figure 2 (b).

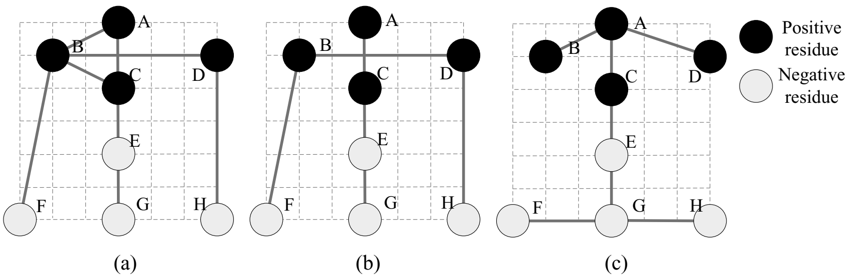


Figure 2. Branch cuts by different methods. (a) from the Goldstein method, (b) from the SMSTA method, (c) from the proposed DBCA method

3.3. the proposed dendriform branch cut algorithm (DBCA)

During the simple spanning tree algorithm, the topological structure of a tree is fixed after the cumulative charge is equal to zero, and we do not consider the distribution of other residues around any more. For example, in Figure 2(b), when it is scanning to residue A, the nearest residue C will be added to the current spanning tree as a child of node A. Then we select the residue E as the nearest unvisited node to residue A and C, and add it to the current spanning tree as a child of node C. We continue these steps until the current spanning tree is balanced, and a tree composed of nodes A, C, E and G is generated. When it is scanning to residue B, the nearest unvisited residue D is detected, but from Figure 2(b) we can see that residues A and C are the nearest residues to residue B. We discard the connection to nearest residue and select the connection to the far residue using simple minimum spanning tree, which makes the branch cuts redundant and intersectant.

We improve the Step2 of the simple minimum spanning tree algorithm for those reasons. During the process of scanning, the newly found residue does not add to the global tree as its child immediately. After found the unvisited residue, we search the nearest residue to the current residue, if the nearest residue is already visited, we set the nearest node as the parent node of the current node and the root of the nearest node as the root of the newly spanning tree, then run the minimum spanning tree algorithm to generate the newly spanning tree. Before running the minimum spanning tree, we should carefully determine the initialized cumulative charge of the current spanning tree. If the ground node is not included in the current spanning tree, we set the charge of the newly added node as the initialized cumulative charge of the current tree; If the ground node is included, we should first remove the ground node from the current tree and compute the cumulative charge of the current spanning tree as its newly initialized cumulative charge. Residues distributed as in Figure 2(a) are connected as in Figure 2(c), and we can see that the length of branch cuts is shortened and intersection is disappeared. The proposed method is easy to implement because the optimized first child, next sibling storing structure is used.

4. Results of experiment and analysis

The hardware configuration of the testing system is: CPU: P4 2.4G; MEMORY: 2G; and operation system is windows XP. The filtered interferogram of Mt.Etna Italy from X-SAR, shown as in Figure 3(a), is used to illustrate the proposed method. The number of residues is 3687 and 1637 before dipole-removing [10] and after the removal respectively. The interferogram is processed by Goldstein's algorithm (Goldstein), simple minimum spanning tree algorithm (SMSTA) and dendriform branch cut algorithm (DBCA). We evaluate the performance of different methods from the length of branch cuts, the number of discontinuous blocks and the execution time of phase unwrapping. The results are shown in Table 1. We can see that the length of the branch cuts and the number of discontinuous blocks both reduce using the spanning tree methods. After detailed analysis of the branch cuts, we conclude that the size of the small discontinuous blocks is limited in one or two pixels due to the width of the branch cuts itself. Although there are a few discontinuous blocks, but have little influence on the whole unwrapping result, and we can remove it by the information around.

Table 1. Statistical result of different unwrapping methods

Method	Length of branch cuts (Pixel)	Number of discontinuous blocks	Execution time (Second)
Goldstein	11019	471	0.719
SMSTA	5729	8	1.888
DBCA	5334	5	1.802

The unwrapping results from the three methods are shown in Figure 3. The performance of Goldstein method is the worst shown in Figure 3(b), so it is seldom used directly in practice. The total length of the branch cuts is 11019 pixels, and the branch cuts are long and intersectant, which lead to the big enclosed areas that can't be unwrapped correctly. The branch cut shown in Figure 3(c) is generated by the SMSTA. Compared with the Goldstein method, the number of enclosed areas is reduced sharply and the total length of the branch cuts also reduced to 5729.

However the connection between the residues is still not the best, and long branch cuts make the unwrapping result discontinuous in the rectangle displayed in Figure 3(c). The branch cuts shown in Figure 3(d) are generated by the proposed DBCA. Compared with Figure 3(b), without long branch cut is the most attractive improvement and the length of the total branch cuts is reduced to 5334. The unwrapping result shown in Figure 3(d) is smooth and ideal using the proposed DBCA, which also removes the 'dead area' formed by the classic branch cut algorithm.

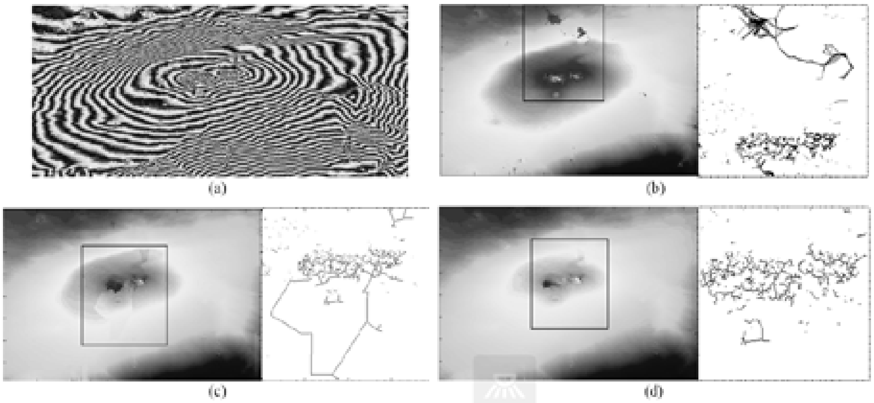


Figure 3. Unwrapping result and branch cuts. (a) Interferogram, (b) from the Goldstein method, (c) from the SMSTA method, (c) from the proposed DBCA method

5. Conclusion

In this paper, a dendriform branch cut algorithm is proposed by combining the classic branch cut algorithm with minimum spanning tree algorithm. In the proposed method, the fist child, next sibling storing structure is adopted to express the residues connective relationship. After establishing the dendriform branch cut algorithm, a branch cut regrowing strategy is proposed, which greatly improves the unwrapping result and removes the 'dead area' formed by the classic branch cut algorithm. The real InSAR dataset is used to validate the performance of the proposed method.

References

[1]Goldstein R.M., Zebker H.A., and Werner C.L. Satellite radar interferometry: two-dimensional phase unwrapping. *Radio Science*, 1988 , pp. 713-720.

[2] Wei Xu, Cumming I. A region growing algorithm for InSAR phase unwrapping. *International Geoscience and Remote Sensing Symposium*, May 1996, pp. 2044-2046.

[3] Pritt M.D. Phase unwrapping by means of multigrid techniques for interferometric SAR. *IEEE Transactions on Geoscience and Remote Sensing*, May 1996, pp. 728-738.

[4] Costantini M. A novel phase unwrapping method based on network programming. *IEEE Transactions on Geoscience and Remote Sensing*, May 1998, pp. 813-821.

[5] Dias J..M.B., Leitao J.M.N. The Z π M algorithm: a method for interferometric image reconstruction in SAR/SAS. *IEEE Transactions on Image Processing*, 2002, pp. 408-422.

[6] Pascazio V., Schirinzi G. A statistical technique for phase unwrapping: application to InSAR data. *Proceedings 2001 International Conference on Image Processing*, Oct 2001, pp. 620-623.

[7] Yang L., Feng Q. and Wang Z.Q. Optimized minimum spanning tree phase unwrapping algorithm for phase image of interferometric SAR. *2006 6th International Conference on ITS Telecommunications*, pp. 1240-1243.

[8] K. Itoh. Analysis of the phase unwrapping algorithm. *Applied Optics*, 1982, pp. 2740.

[9] Ghiglia D.C., Pritt M.D. *two-dimensional phase unwrapping: theory, algorithm, and software.* New York: John Wiley & sonship, Inc, 1998.

[10] Huntley J.M., Noise-immune phase unwrapping algorithm. Applied Optics, 1989, pp. 3268-3270.